Anticollusion solutions for asymmetric fingerprinting protocols based on client side embedding

Bianchi *et al. EURASIP Journal on Information Security* (2015) 2015:6
DOI 10.1186/s13635-015-0023-y

EURASIP Journal on Information Security
a SpringerOpen Journal

RESEARCH Open Access

Anticollusion solutions for asymmetric fingerprinting protocols based on client side embedding

Tiziano Bianchi[1*], Alessandro Piva[2] and Dasara Shullani[2]

Abstract

In this paper, we propose two different solutions for making a recently proposed asymmetric fingerprinting protocol based on client-side embedding robust to collusion attacks. The first solution is based on projecting a client-owned random fingerprint, securely obtained through existing cryptographic protocols, using for each client a different random matrix generated by the server. The second solution consists in assigning to each client a Tardos code, which can be done using existing asymmetric protocols, and modulating such codes using a specially designed random matrix. Suitable accusation strategies are proposed for both solutions, and their performance under the averaging attack followed by the addition of Gaussian noise is analytically derived. Experimental results show that the analytical model accurately predicts the performance of a realistic system. Moreover, the results also show that the solution based on independent random projections outperforms the solution based on Tardos codes, for different choices of parameters and under different attack models.

Keywords: Fingerprinting; Anticollusion codes; Buyer-seller watermarking protocol; Client-side embedding; Secure watermark embedding

Introduction

The wide availability of platforms for the distribution of multimedia contents poses several problems regarding copyright protection. A possible solution to prevent copyright violations is to embed a unique code, or fingerprint, in the distributed copies, so that illegally redistributed copies could be traced back to the entity responsible for the violation. In the literature, several watermarking techniques have been proposed with the aim of embedding a unique fingerprint in a multimedia content [1, 2]. However, for the practical deployment of such systems, a number of problems have to be solved.

First, the actual fingerprint can not be embedded by the distributor alone, since a guilty customer could claim to have been framed by a malicious distributor, undermining the credibility of the whole system. Hence, fingerprinting must be either handled by a trusted third party, or performed interactively via secure cryptographic protocols [3–6]. Second, if individual fingerprinted copies for each customer are generated at the server side, the amount of computational and bandwidth resources needed by the server may soon become prohibitively high in large-scale systems. A solution to this issue is offered by client-side embedding methods, in which the server distributes the same encrypted copy of the content to all the clients, along with different client-specific decryption keys allowing each user to decrypt a slightly different version of the content, bearing a different watermark [7–10]. An alternative solution consists in creating a few streams, each with different but constant watermarks, and force the client to switch between the streams [11].

Besides these first two issues, a practical system has also to cope with malicious customers trying to attack the system. In this sense, a very effective attack is the so called *collusion* attack. In this attack, a coalition of customers combine their differently watermarked copies in order to obtain a new copy in which the watermark is much

*Correspondence: tiziano.bianchi@polito.it
[1] Department of Electronics and Telecommunications, Politecnico di Torino, Corso Duca Degli Abruzzi, 24, 10129 Torino, Italy
Full list of author information is available at the end of the article

harder to be detected. As to multimedia fingerprinting, several studies have analyzed the robustness of watermarking techniques to collusion attacks [12, 13], the effectiveness of different detectors [14], and the robustness to different collusion strategies [15]. An alternative solution is to construct specific anticollusion codes that can be embedded in a content [16]. Asymptotically optimal probabilistic anticollusion codes were first proposed by Tardos in [17], and later optimized under several aspects [18–31].

The current literature on fingerprinting usually addresses only a single issue at a time. There are few solutions that try to simultaneously solve the above issues. In [32], we proposed an asymmetric fingerprint protocol based on a client-side distribution paradigm. The above solution effectively solves both customer's rights and scalability problems: nevertheless, collusion resistance was not addressed in this proposal. Recently, a few works have investigated the use of anticollusion codes in asymmetric fingerprinting protocol. In [33], the authors propose an asymmetric Tardos code construction based on oblivious transfer. In [34], the author proposes an asymmetric binary fingerprinting code based on Boneh-Shaw codes. However, both solutions are based on a server-side distribution framework. Some authors also investigated the use of Tardos codes in a client-side distribution framework [35], but without addressing asymmetric protocols.

In this paper, we propose two strategies for providing collusion resistance in the protocol of [32]. The first strategy is based on generating independent coding matrices for the fingerprint of different users, and is conceptually similar to using near orthogonal independent Gaussian fingerprints. The second strategy consists in generating the fingerprint of each user according to a Tardos code, exploiting the fact that such codes can be securely distributed using the protocol proposed in [33]. Since Tardos codes are much longer than the random fingerprints used in [32], an efficient encoding obtained through a random partially circulant matrix is proposed, which is conceptually similar to dimensionality reduction techniques based on the Johnson-Lindenstrauss lemma [36] and applied in compressed sensing [37]. For both solutions, we develop the corresponding accusation strategies and we analytically derive their anticollusion performance under the averaging attack. Experimental results are finally presented to support the proposed analysis.

Background

In this section, we briefly review the client-side embedding technique proposed in [38] and the asymmetric fingerprinting protocol proposed in [32], which builds on the above technique.

LUT-based client-side embedding

The client-side embedding proposed by Celik et al. in [8, 38] is based on a long-term master encryption look-up table $\mathbf{E}$ of size T and a set of watermarking LUTs $\mathbf{W}_k$ of the same size, $k = 0, \ldots, N_U - 1$, each associated to one of the N_U clients. The entries of $\mathbf{E}$ are usually i.i.d. random variables following a Gaussian distribution with variance σ_E^2, while the entries of each $\mathbf{W}_k$ are i.i.d. random variables following a Gaussian distribution with variance σ_W^2. Different LUTs are assumed to be independent. For the kth client, the distribution server generates a personalized decryption LUT $\mathbf{D}_k$ by combining componentwise the master encryption LUT $\mathbf{E}$ and a watermark LUT $\mathbf{W}_k$ as

$$\mathbf{D}_k(t) = -\mathbf{E}(t) + \mathbf{W}_k(t) \tag{1}$$

for $t = 0, 1, \ldots, T-1$. The personalized decryption LUTs are then transmitted once to each client over a secure channel. Note that the generation of the LUTs is carried out just once at the setup phase.

For the secure distribution of a content, a set of $M \times R$ values t_{ih} in the range $[\,0, T-1]$, where $0 \leq i \leq M-1, 0 \leq h \leq R-1$, is pseudo-randomly generated according to a content dependent key *sek*. Each of the M content features x_i is encrypted by adding R entries of the encryption LUT identified by the indexes $(t_{i0}, \ldots, t_{i(R-1)})$, obtaining the encrypted feature c_i as follows

$$c_i = x_i + \sum_{h=0}^{R-1} \mathbf{E}(t_{ih}). \tag{2}$$

Joint decryption and watermarking is performed by generating the same sequence of indexes t_{ih} according to the content dependent key *sek* and by adding R entries of the decryption LUT $\mathbf{D}_k$ to each encrypted feature c_i as

$$y_{k,i} = c_i + \sum_{h=0}^{R-1} \mathbf{D}_k(t_{ih}) = x_i + \sum_{h=0}^{R-1} \mathbf{W}_k(t_{ih}) = x_i + w_{k,i} \tag{3}$$

where the ith watermark component is given as the sum of R entries of the LUT $\mathbf{W}_k$. The result of this operation is the watermarked content $\mathbf{y}_k = \mathbf{x} + \mathbf{w}_k$ identifying the kth user.

Asymmetric fingerprinting

A typical asymmetric fingerprinting protocol [3] is composed of a registration phase, in which the client proves his/her identity and commits to a secret that only he/she knows, and a watermarking protocol, jointly performed by the client and the distribution server, after which only the client receives a copy of the watermarked content containing his/her secret. If the copy is illegally distributed, the server can identify the guilty client by extracting his/her secret from the watermark and prove to a Judge that it

is indeed the client's secret by using a proper dispute resolution protocol based on the client's commitment.

Let us assume that each client has a public/private key pair (puk, prk). The schemes proposed in [5, 6] assume that the kth client produces as input to the watermarking protocol a random string of L bits denoted as $\mathbf{b}_k$, representing his/her secret. This L-bit fingerprint is encrypted with the client's public key using an additively homomorphic cryptosystem and sent to the server, together with a proper commitment linking the identity of the client to the encrypted fingerprint. Thanks to the homomorphic properties of the encryption, the server is able to compute a watermarked copy of the content containing the client's fingerprint directly in the encrypted domain, so that the client's secret is never disclosed. However, if after an illegal redistribution a watermarked copy is found in the clear, the client's secret can be obtained through the watermark decoder and used in an accusation protocol.

In order to use an asymmetric fingerprinting protocol within a client-side distribution framework, the authors of [32] propose to use the above protocol to securely embed the client's secret in the encryption LUT and to employ the resulting modified LUT as the client's decryption LUT. Namely, the fingerprint of the kth client is first randomized by the server using a secret L-bit sequence $\mathbf{r}_k$. Then, the resulting string is encoded using a binary antipodal modulation, yielding the to be transmitted message $\mathbf{m}_k$. Formally, each symbol of the message is computed as $m_{k,l} = \sigma_W(2(b_{k,l} \oplus r_{k,l}) - 1)$, $0 \leq l \leq L-1$. Then, the message $\mathbf{m}_k$ is projected according to a $T \times L$ random matrix $\mathbb{G}$, yielding the following watermarking LUT

$$\mathbf{W}_k = \mathbb{G}\mathbf{m}_k \tag{4}$$

The personalized decryption LUT $\mathbf{D}_k$ is finally obtained by combining the encryption LUT and the above watermarking LUT as

$$\mathbf{D}_k = -\mathbf{E} + \mathbb{G}\mathbf{m}_k. \tag{5}$$

Since all the above operations are linear, the encryption of $\mathbf{D}_k$ can be directly computed in the encrypted domain using the encryption of $\mathbf{b}_k$, obtained through an additively homomorphic cryptosystem, and properly rescaled and quantized versions of $\mathbf{E}$ and $\mathbb{G}$ [32]. A high-level block diagram of the above protocol is shown in Fig. 1.

The watermarked content obtained after the decryption with the above $\mathbf{D}_k$ can be expressed by adding to the encrypted signal the product of the decryption LUT $\mathbf{D}$ and a proper binary matrix $\mathbb{T}$ defined according to the sequence of indexes t_{ih}, i.e.,

$$\mathbf{y} = \mathbf{c} + \mathbb{T}\mathbf{D}_k = \mathbf{x} + \mathbb{T}\mathbf{W}_k \tag{6}$$

where $\mathbb{T}$ is a $M \times T$ binary matrix defined as

$$\mathbb{T}(i,j) = \begin{cases} 1 \;\; j = t_{ih},\; h = 0, \ldots, R-1 \\ 0 \;\; \text{otherwise.} \end{cases} \tag{7}$$

If we assume that the watermark decoder receives a copy of the watermarked signal corrupted by an additive noise, the received signal can be expressed as a function of the client's modulated fingerprint as

$$\mathbf{y}' = \mathbf{y} + \mathbf{n} = \mathbf{x} + \mathbb{T}\mathbb{G}\mathbf{m}_k + \mathbf{n} = \mathbf{x} + \tilde{\mathbb{G}}\mathbf{m}_k + \mathbf{n} \tag{8}$$

that is, the actual watermark is equal to the message $\mathbf{m}_k$ projected by the $M \times L$ random matrix $\tilde{\mathbb{G}} = \mathbb{T}\mathbb{G}$.

Since the scheme is asymmetric, the decoder does not know the messages $\mathbf{m}_k$ and a correlation detector, as that proposed in [8], is not applicable here. Hence, the authors in [32] propose to obtain an estimated fingerprint $\hat{\mathbf{b}}_k$ and to verify whether it matches with a recorded client, using the accusation protocols provided by the underlying asymmetric fingerprinting protocol.

When the original signal $\mathbf{x}$ is available at the decoder, which is a common hypothesis in fingerprint applications, a simple decoder can be obtained using the *Matched Filter (MF)* principle as

$$\hat{\mathbf{b}}_k = \text{sgn}\left\{\tilde{\mathbb{G}}^T(\mathbf{y}' - \mathbf{x})\right\} \tag{9}$$

where sgn {} denotes the sign function.

Anticollusion solutions

As pointed out in [32], the basic protocol described in the previous section is potentially vulnerable to collusion attacks, where several clients combine their received watermarked copies in order to obtain a pirated copy in which their fingerprints are much harder to be detected. In the following, we introduce two possible solutions to cope with this problem. The first solution is based on the use of random fingerprints modulated by a different projection matrix for each client. The second solution consists in generating each client's fingerprint according to a Tardos anticollusion code.

Collusion model

We assume that a server distributes N_U differently watermarked copies of the same content $\mathbf{x}$ to N_U clients. Among those clients, a coalition $\mathcal{C}$ of $C = |\mathcal{C}|$ clients cooperates in order to obtain a pirated copy $\mathbf{y}_C$ according to some, possibly randomized, collusion strategy. In the following, we assume that the C colluders adopt an average collusion strategy, i.e., the pirated copy is obtained as

$$\mathbf{y}_C = \frac{1}{C}\sum_{c \in \mathcal{C}} \mathbf{y}_c + \mathbf{n} \tag{10}$$

where $\mathbf{y}_k$ denotes the watermarked copy of the kth client and $\mathbf{n}$ is additive Gaussian noise with zero mean, independent of the watermarked copies. In multimedia

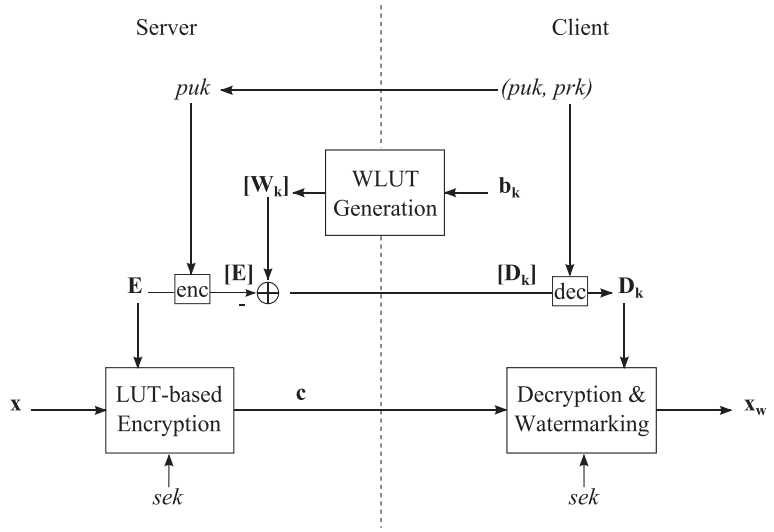


Fig. 1 Block diagram of asymmetric fingerprinting based on client side embedding

fingerprinting, for a given bound on the distortion of $\mathbf{y}_C$ with respect to the watermarked copies, such a strategy is the most efficient from the attacker's point of view [14]. In the client-side setting, a coalition of attackers has the option of directly combining the respective LUTs, instead of the content. As to the averaging attack, in the absence of noise, the two strategies are equivalent. In the presence of noise, attacking the LUT may achieve a lower distortion of the content for the same variance of the additive noise.

Given a suspect content $\mathbf{y}_C$, we assume that a detector produces an accusation set $\mathcal{A}$ and accuses the kth client if $k \in \mathcal{A}$. The performance of the detector is measured by the probability of accusing at least one colluder, referred to as *probability of detection*, and by the probability of accusing an innocent user, referred to as *probability of false alarm*. Formally, the probability of detection is given by

$$P_d = Pr\{\mathcal{A} \cap \mathcal{C} \neq \emptyset\} \tag{11}$$

whereas the probability of false alarm is given by

$$P_{fa} = Pr\left\{\mathcal{A} \cap \overline{\mathcal{C}} \neq \emptyset\right\}. \tag{12}$$

If we assume that the probability of accusing a specific innocent client is ϵ, from the union bound we have that $P_{fa} \leq (N_U - C)\epsilon < N_U\epsilon$. A fingerprinting system is said to be secure against a coalition of C colluders for a probability of false alarm η if the corresponding probabilities of detection and false alarm satisfy $P_d \approx 1$ and $P_{fa} \leq \eta$. In practice, $P_d \geq 0.9$ is usually sufficient to deter collusion.

Random LUTs

Since the early days of watermarking, it was argued that additive Gaussian distributed watermarks may provide a certain level of resistance to colluders [12]. In [13], it was shown that a system with N_U users employing Gaussian watermarks of length M, i.e., where the fingerprints are represented by vectors of M i.i.d. zero-mean Gaussian variables, is secure against a coalition of C colluders, where $C = O\left(\sqrt{M/\log N_U}\right)$, which is similar to the asymptotic behavior of Tardos codes [17]. If we assume that the random projection matrix $\mathbb{G}$ is composed of i.i.d. Gaussian variables, then the watermarking technique described by Eqs. (5) and (6) is equivalent to adding to the content L almost independent Gaussian watermarks of length M, each modulated by a fingerprint bit. Hence, by assigning to different clients different and independent projection matrices, we expect that the system will benefit from the inherent anticollusion properties of Gaussian watermarks.

In the proposed solution, we assume that each client is identified by a random L-bit fingerprint, where each bit is independently drawn with probability 0.5. The fingerprint of the kth client is then randomized using a server generated bit sequence $\mathbf{r}_k$ and encoded using binary antipodal modulation in a L-symbol vector $\mathbf{m}_k$ and the server distributes to each client a personalized decryption LUT obtained as

$$\mathbf{D}_k = -\mathbf{E} + \mathbb{G}_k\mathbf{m}_k. \tag{13}$$

where $\mathbb{G}_k$ is the random projection matrix associated to the kth client. When the server receives a suspect copy $\mathbf{y}_C$,

an estimate of the fingerprint is computed for each client according to

$$\hat{\mathbf{b}}_k = \text{sgn}\left\{\tilde{\mathbb{G}}_k^T(\mathbf{y}_C - \mathbf{x})\right\} \oplus \mathbf{r}_k \tag{14}$$

where $\tilde{\mathbb{G}}_k = \mathbb{T}\mathbb{G}_k$ and compared with the recorded fingerprint of the corresponding kth client. Namely, for each client the detector computes the Hamming distance between the estimated and the recorded fingerprint, producing the accusation score

$$Z_k = \sum_{l=0}^{L-1}\left(\hat{b}_{k,l} \oplus b_{k,l}\right) \tag{15}$$

and declares as guilty the clients belonging to the following set

$$\mathcal{A} = \{k | Z_k < \theta_R\}. \tag{16}$$

The threshold θ_R must be set in order to satisfy a suitable bound on the probability of accusing innocent clients. In order to guarantee $P_{fa} \leq \eta$, a conservative choice is to impose $Pr\{Z_k < \theta_R\} \leq \eta/N_U$ for each $k \notin \mathcal{C}$. When $k \notin \mathcal{C}$, we have that $\tilde{\mathbb{G}}_k$ and $\mathbf{y}_C$ are independent, and the bits $\hat{b}_{k,l}$ can be modeled as Bernoulli variables with probability 0.5 independent of the fingerprint bits $b_{k,l}$. As a consequence, the accusation score of an innocent user is distributed according to a binomial distribution $B(L, 0.5)$ and the probability $Pr\{Z_k < \theta_R | k \notin \mathcal{C}\}$ can be upper bounded using Hoeffding's inequality as

$$Pr\{Z_k < \theta_R | k \notin \mathcal{C}\} \leq \exp\left(-\frac{2(L/2 - \theta_R)^2}{L}\right) \tag{17}$$

from which we derive the threshold as

$$\theta_R = \frac{L}{2} - \sqrt{\frac{L\log(N_U/\eta)}{2}}. \tag{18}$$

For large L, a less conservative threshold can be obtained by approximating Z_k as a Gaussian distributed variable with mean $L/2$ and variance $L/4$, which results in the threshold

$$\theta_R = \frac{L}{2} + \frac{\sqrt{L}}{2}\Phi^{-1}(\eta/N_U) \tag{19}$$

where Φ^{-1} denotes the inverse cumulative distribution function of a standard normal variable.

Tardos codes

An alternative solution for providing resistance to collusion in the protocol described in the previous section is to construct the clients' fingerprints $\mathbf{b}_k$ according to some anticollusion code. Tardos in [17] proposed a probabilistic code construction which is asymptotically optimal under the restricted digit model, i.e., under the assumption that the attackers can only use one of the symbols that they have received at each position to produce the colluded copy. The construction of the code is as follows. Given a fingerprint length L and a maximum coalition size C, the server generates the random numbers $p_l \in [\delta, 1-\delta]$, $l = 0, \ldots, L-1$ according to the probability density function $f(p) = \left(\left(\pi - 4\arcsin\sqrt{\delta}\right)\sqrt{p(1-p)}\right)^{-1}$, where the parameter δ satisfies $\delta C \ll 1$ (Tardos suggests $\delta = 1/(300C)$). Then, for each client, the fingerprint bits $b_{k,l}$, $k = 0, \ldots, N_U - 1$, are randomly generated as Bernoulli random variables with $Pr\{b_{k,l} = 1\} = p_l$.

In the proposed scheme, we assume that Tardos codes can be securely distributed according to an asymmetric protocol and embedded at the Server's side using a homomorphic cryptosystem, for example using the solutions proposed in [33]. Following the approach in [32], we assume that a similar technique can be used for securely embedding a Tardos code in a LUT. However, a strategy allowing to embed a sufficiently long Tardos code in a watermarking LUT, and next to detect a possibly pirated codeword usable as input to the accusation protocol, has to be designed.

In [19, 30], the authors show that the optimal length of a Tardos code that is secure against a coalition of C colluders, when using the symmetric accusation score, is $L \approx \frac{1}{2}\pi^2 C^2 \log(N_U/\eta)$, which means that practical code lengths are of the order of $L \approx 10^6$. As a consequence, the computation of the watermarking LUT in (5) may require a very large projection matrix $\mathbb{G}$. Moreover, in some cases, we may have $L > M$, which implies that a zero forcing detection approach as that proposed in [32] is not possible, since $\tilde{\mathbb{G}}^T\tilde{\mathbb{G}}$ becomes a singular matrix.

In order to manage large values of L, we propose thus to generate $\mathbb{G}$ according to a partial random circulant matrix. A random circulant matrix is based on a random vector that is circularly shifted to generate every row. Moreover, circulant matrices can be diagonalized using a discrete Fourier transform (DFT) as

$$\mathbb{G} = \mathbb{W}^H\mathbb{D}\mathbb{W} \tag{20}$$

where $\mathbb{W}$ is the unitary DFT matrix and $\mathbb{D}$ is a diagonal matrix whose nonzero elements are the DFT of the first column of $\mathbb{G}$. The above property means that the watermarking LUT can be efficiently computed by relying on a fast Fourier transform. For $L > M$, the watermarking LUT can be computed as

$$\mathbf{W}_k = \mathbb{P}\mathbb{W}^H\mathbb{D}\mathbb{W}\mathbf{m}_k \tag{21}$$

where the $M \times L$ matrix $\mathbb{P}$ selects the first M entries of a vector of length L, whereas for $L \leq M$ the watermarking LUT can be computed as

$$\mathbf{W}_k = \mathbb{W}^H\mathbb{D}\mathbb{W}\mathbb{P}^T\mathbf{m}_k \tag{22}$$

where the $M \times L$ matrix $\mathbb{P}^T$ pads a L-length vector with $M - L$ zeros. We note that the above expressions can

be also implemented with an encrypted version of $\mathbf{m}_k$ by using an encrypted domain FFT algorithm like the one proposed in [39].

As to the detection of the fingerprint, the matched filter detector in (9) can be computed on a suspect copy $\mathbf{y}_C$ as

$$\hat{\mathbf{b}} = \text{sgn}\left\{\mathbb{W}^H\mathbb{D}^H\mathbb{W}\mathbb{P}^T\mathbb{T}^T(\mathbf{y}_C - \mathbf{x})\right\} \tag{23}$$

for $L > M$, or as

$$\hat{\mathbf{b}} = \text{sgn}\left\{\mathbb{P}\mathbb{W}^H\mathbb{D}^H\mathbb{W}\mathbb{T}^T(\mathbf{y}_C - \mathbf{x})\right\} \tag{24}$$

for $L < M$. In the above expression, we note that computing the matrix product $\mathbb{T}^T(\mathbf{y}_C - \mathbf{x})$ is actually equivalent to constructing a suspect LUT from the sequence of watermark estimates as explained in [8, Sec. IV.B] and does not require an actual matrix multiplication. Once we have an estimate of the suspect fingerprint, different strategies exist in order to detect possible colluders.

The first solution we employ in this paper is the symmetric version of the accusation sums proposed in [19]. Namely, for each client the detector computes the following score

$$Z_k = \sum_{l=0}^{L-1}\left(2\hat{b}_l - 1\right)g_{b_{k,l}}(p_l) \tag{25}$$

where the weights are defined as $g_1(p_l) = \sqrt{(1-p_l)/p_l}$ and $g_0(p_l) = -1/g_1(p_l)$, and declares as guilty the clients belonging to the following set

$$\mathcal{A} = \{k|Z_k > \theta_T\}. \tag{26}$$

In [19], it is shown that in order to guarantee $P_{fa} \leq \eta$ the threshold must be chosen as

$$\theta_T = 2\sqrt{L\left\lceil\log(N_U/\eta\right\rceil}. \tag{27}$$

A less conservative threshold can be also obtained by using the central limit theorem. In this case, the accusation sum for innocent clients can be approximated as a Gaussian variable with zero mean and variance L [19], which results in the threshold

$$\theta_T = \sqrt{L}\Phi^{-1}(1-\eta/N_U). \tag{28}$$

The second solution is a detector optimized for the asymptotic worst-case attack on Tardos codes, which is the so-called interleaving attack [40]. In order to model the noise at the receiver, let us assume that each fingerprint bit is received through a binary symmetric channel (BSC) with crossover probability p_e. Using the results in [29] for the combined digit model, the optimal score in the case of the interleaving attack can be expressed as

$$Z_k = \sum_{l=0}^{L-1} h_{b_{k,l},\hat{b}_l}(p_l) \tag{29}$$

where the optimal suspicion function h is defined as

$$h_{0,0}(p) = p/(1-p+\gamma_e) \tag{30}$$

$$h_{0,1}(p) = -p/(p+\gamma_e) \tag{31}$$

$$h_{1,0}(p) = -(1-p)/(1-p+\gamma_e) \tag{32}$$

$$h_{1,1}(p) = (1-p)/(p+\gamma_e) \tag{33}$$

with $\gamma_e = p_e/(1-2p_e)$. The detector declares as guilty the clients belonging to the set

$$\mathcal{A} = \{k|Z_k > \theta_{T2}\} \tag{34}$$

where θ_{T2} should be computed taking into account the distribution of the accusation sum for innocent clients. It is easy to verify $E[\,Z_k|k \notin \mathcal{C}] = 0$, whereas

$$\begin{aligned} &Var[Z_k|k \notin \mathcal{C}] \\ &= L \cdot E_p\left[\frac{p(1-p)}{(1-p+\gamma_e)^2}P_{\hat{b}_l}(0) + \frac{p(1-p)}{(p+\gamma_e)^2}P_{\hat{b}_l}(1)\right] \\ &= \sigma^2_{T2,I}. \end{aligned} \tag{35}$$

Hence, using a Gaussian approximation, the threshold can be derived as

$$\theta_{T2} = \sigma_{T2,I}\Phi^{-1}(1-\eta/N_U). \tag{36}$$

Differently from the symmetric accusation score in (25), the threshold for the optimized accusation score depends on the number of colluders, their attack strategy, and the noise level at the receiver. Since these quantities are usually unknown to the detector, a practical detector should estimate $\sigma^2_{T2,I}$ from the received fingerprint in order to compute the right threshold.

Practical issues

In the secure implementation of the asymmetric fingerprinting protocol of [32], the server does not have the plaintext values of the fingerprints $\mathbf{b}_k$ provided by each kth client. Hence, after decoding either the suspect fingerprint of each kth client with (14) or decoding the global suspect fingerprint with (23) or (24), a secure protocol must be invoked to compare such fingerprint estimates with the actual fingerprints of each user.

As to the random LUT solution, the accusation set defined in (16) can be securely computed by using secure Hamming distance protocols, like those described in [41]. As to the Tardos solution, after the fingerprint bits are demodulated using (9), the scores in (25) can be computed by using the accusation procedure described in [33], which requires a trusted Judge to be revealed both the random bias values p_l, which are a secret input of the server, and the fingerprints bits $b_{k,l}$, which are a secret input of the kth client. In the first case, the computation of the accusation sets requires the execution of

onerous cryptographic protocols, while in the second case, a trusted third party is required.

However, when the number of active clients is very large, e.g., millions of users, performing the secure computation of the accusation sets may become prohibitively expensive. A possible solution to this issue is to embed a server specific watermark along with the client secret watermark and using this first watermark to compute a set of suspect clients, as suggested in [33]. Since this watermark is known to the server, it can not be used in a dispute resolution protocol as a proof that a given client is guilty. However, once the server has a list of suspect clients, the secure accusation protocols can be run only on this reduced subset. Typically, the list of suspect clients will only contain few tens of entries, so that the previously described secure protocols become feasible.

In order to embed a server specific watermark in a client-side distribution framework, the server can simply add to each personalized decryption LUT $\mathbf{D}_k$ a server specific watermarking LUT $\mathbf{SW}_k$, i.e., the actual decryption LUT can be redefined as

$$\mathbf{D}_k = -\mathbf{E} + \mathbf{W}_k + \mathbf{SW}_k. \tag{37}$$

By using the above decryption LUT, each client will embed along with his/her secret fingerprint a server-specific spread spectrum watermark [8], which is known to have good anticollusion properties. Moreover, the above operation can be performed in the encrypted domain by relying on a homomorphic cryptosystem, hence the secure distribution of decryption LUTs is not violated. A possible drawback is that the two watermarks will interfere each other. However, this will likely affect only the performance of the server specific watermarks, since the watermarks of the suspect clients can be subtracted from the content under investigation before computing the final accusation sets.

Performance analysis

In this section, we will provide analytical expressions for the probability of detection of the proposed anticollusion schemes as a function of the number of colluders and the probability of false alarm. The analysis is based on the simplifying assumption that the final projection matrix $\tilde{\mathbb{G}}$ is made of i.i.d. Gaussian entries and relies heavily on the central limit theorem, so it is valid only for large M and L.

Random LUTs

From (10) and the definition of watermarking LUT in (13), we can express the estimated watermark after an average attack by C colluders as

$$\mathbf{w} = \mathbf{y}_C - \mathbf{x} = \frac{1}{C}\sum_{c\in\mathcal{C}} \tilde{\mathbb{G}}_c \mathbf{m}_c \tag{38}$$

Let us express the lth column of $\tilde{\mathbb{G}}_c$ as $\boldsymbol{\phi}_{c,l}$. We have

$$\mathbf{w} = \frac{1}{C}\sum_{c\in\mathcal{C}}\sum_{l=0}^{L-1} m_{c,l}\boldsymbol{\phi}_{c,l}. \tag{39}$$

According to the MF detector, the lth detected symbol for the kth client is given by $\hat{m}_{k,l} = \boldsymbol{\phi}_{k,l}^T\mathbf{w}$. Now, let us assume that the vectors $\boldsymbol{\phi}_{c,l}$ are mutually independent and Gaussian distributed, and that they are normalized so that the variance of each component is equal to $1/(ML)$. By invoking the central limit theorem, we can model $\hat{m}_{k,l}$, both when $k \in \mathcal{C}$ and $k \notin \mathcal{C}$, as a Gaussian distributed variable. Moreover, it is easy to derive the following statistics

$$E[\hat{m}_{k,l}|k\in\mathcal{C}] = m_{k,l}\cdot\frac{1}{CL} = m_{k,l}\mu_C \tag{40}$$

$$Var[\hat{m}_{k,l}|k\in\mathcal{C}] = \frac{CL+1}{C^2L^2M} + \frac{\sigma_n^2}{L} = \sigma_C^2 \tag{41}$$

where σ_n^2 denotes the variance of the components of the noise vector $\mathbf{n}$. Hence, we can express the probability of reading the wrong bit for a colluder as

$$p_e = Pr\left\{\text{sgn}\{\hat{m}_{k,l}\} \neq b_{k,l}\right\} = Q\left(\frac{\mu_C}{\sigma_C}\right) \tag{42}$$

where $Q(x) = (2\pi)^{-1/2}\int_x^\infty e^{-t^2/2}dt$ is the tail probability of a standard normal variable. By assuming that different fingerprint symbols are independent, the accusation score in (16) can be modeled by a binomial $B(L, p_e)$ distribution. Using the Gaussian approximation of the binomial distribution, the probability of detecting a specific colluder for a threshold θ_R can be expressed as

$$P_{d,\text{single}} = 1 - Q\left(\frac{\theta_R - Lp_e}{\sqrt{Lp_e(1-p_e)}}\right) \tag{43}$$

and, assuming independence for different colluders, the probability of detecting at least a colluder is estimated as

$$P_d = 1 - (1 - P_{d,\text{single}})^C. \tag{44}$$

Tardos codes

In [18], it was shown that the performance of Tardos codes with the accusation scores as in (25) can be analyzed using a Gaussian approximation. Let us define $\mu_T = E[Z_k|k\in\mathcal{C}]$ and $\sigma_T^2 = Var[Z_k|k\in\mathcal{C}]$, denoting the mean and the variance of the accusation score for a generic colluder, respectively. The probability

of detecting a specific colluder for a threshold θ_T can be expressed as

$$P_{d,\text{single}} = 1 - Q\left(\frac{\mu_T - \theta_T}{\sigma_T}\right) \tag{45}$$

whereas, assuming independence for different colluders, the probability of detecting at least a colluder can be estimated as in (44).

For the accusation score in (25), the expectation of Z_k, computed over p, $\hat{\mathbf{b}}$, $\mathbf{b}_k$, can be expressed as

$$\mu_T = \sum_{l=0}^{L-1} E_{p_l}\left[E_{\hat{b}_l,b_{k,l}}\left[(2\hat{b}_l - 1)g_{b_{k,l}}(p_l)\right]\right]. \tag{46}$$

Since the bias terms p_l are i.i.d., this can be simplified as $E[Z_k] = L \cdot E_p\left[E_{\hat{b},b_k}\left[(2\hat{b}-1)g_{b_k}(p)\right]\right]$, where

$$\begin{aligned}
&E_{\hat{b},b_k}\left[(2\hat{b}-1)g_{b_k}(p)\right]\\
&=g_1(p)P_{\hat{b},b_k}(1,1) + g_0(p)P_{\hat{b},b_k}(1,0)\\
&\quad - g_1(p)P_{\hat{b},b_k}(0,1) - g_0(p)P_{\hat{b},b_k}(0,0)\\
&=g_1(p)p\left[P_{\hat{b}|b_k}(1|1) - P_{\hat{b}|b_k}(0|1)\right]\\
&\quad + g_0(p)(1-p)\left[P_{\hat{b}|b_k}(1|0) - P_{\hat{b}|b_k}(0|0)\right]\\
&=g_1(p)p\left[2P_{\hat{b}|b_k}(1|1) - 1\right]\\
&\quad - g_0(p)(1-p)\left[2P_{\hat{b}|b_k}(0|0) - 1\right]\\
&=2\sqrt{p(1-p)}\left[P_{\hat{b}|b_k}(1|1) + P_{\hat{b}|b_k}(0|0) - 1\right].
\end{aligned} \tag{47}$$

The variance of Z_k can be similarly expressed as

$$\begin{aligned}
\sigma_T^2 =& E\left[\left(\sum_{l=0}^{L-1}(2\hat{b}_l - 1)g_{b_{k,l}}(p_l)\right)^2\right] - E[Z_k]^2\\
=& \sum_{l=0}^{L-1} E\left[(2\hat{b}_l - 1)^2 g_{b_{k,l}}^2(p_l)\right]\\
&+ \sum_{l=0}^{L-1}\sum_{l'=0,l'\neq l}^{L-1} E\left[(2\hat{b}_l - 1)g_{b_{k,l}}(p_l)\right]\\
&\times E\left[\left(2\hat{b}_{l'} - 1\right)g_{b_{k,l'}}\left(p_l'\right)\right] - E[Z_k]^2\\
=& \sum_{l=0}^{L-1} E\left[g_{b_{k,l}}^2(p_l)\right] - E[Z_k]^2\\
=& L - E[Z_k]^2
\end{aligned} \tag{48}$$

since we have $E\left[g_{b_{k,l}}^2(p_l)\right] = p_l g_1^2(p_l) + (1-p_l)g_0^2(p_l) = 1$.

For the accusation score in (29), the expectation of Z_k can be computed as $\mu_T = L \cdot E_p\left[E_{\hat{b},b_k}\left[h_{b_k,\hat{b}}(p)\right]\right]$, where

$$\begin{aligned}
&E_{\hat{b},b_k}\left[h_{b_k,\hat{b}}\right]\\
&=h_{1,1}(p)P_{\hat{b},b_k}(1,1) + h_{1,0}(p)P_{\hat{b},b_k}(0,1)\\
&\quad + h_{0,1}(p)P_{\hat{b},b_k}(1,0) + h_{0,0}(p)P_{\hat{b},b_k}(0,0)\\
&=p(1-p)\left[\frac{P_{\hat{b}|b_k}(0|0)}{1-p+\gamma_e} - \frac{1-P_{\hat{b}|b_k}(0|0)}{p+\gamma_e}\right.\\
&\quad \left. + \frac{P_{\hat{b}|b_k}(1|1)}{p+\gamma_e} - \frac{1-P_{\hat{b}|b_k}(1|1)}{1-p+\gamma_e}\right]\\
&=p(1-p)\left[\frac{1+2\gamma_e}{(1-p+\gamma_e)(p+\gamma_e)}\right.\\
&\quad \left. \times \left(P_{\hat{b}|b_k}(1|1) + P_{\hat{b}|b_k}(0|0) - 1\right)\right].
\end{aligned} \tag{49}$$

Similarly, the variance can be computed as $\sigma_T = L \cdot E_p\left[E_{\hat{b},b_k}\left[h_{b_k,\hat{b}}^2(p)\right]\right] - \mu_T^2$.

In order to evaluate μ_T and σ_T for the different accusation scores, the collusion channel $P_{\hat{b}|b_k}(\cdot|\cdot)$ needs to be characterized. In the proposed scheme, Tardos codes are used over a noisy channel, and the random modulation of the code by $\tilde{\mathbb{G}}$ is not perfectly orthogonal, resulting in an additional interference over each code bit. The above model can be analyzed considering a collusion attack in the combined digit model [25]. According to the average collusion model in (10), for each fingerprint position the colluders output a symbol proportional to the number of zeros and ones that they see in their copies. If we express the lth column of $\tilde{\mathbb{G}}$ as $\boldsymbol{\phi}_l$, then we have

$$\mathbf{w} = \sum_{l=0}^{L-1} \boldsymbol{\phi}_l \frac{1}{C}\sum_{c\in\mathcal{C}} m_{c,l} \tag{50}$$

and the lth symbol computed by the MF detector is given by $\hat{m}_l = \boldsymbol{\phi}_l^T \mathbf{w}$. If we assume that in the lth position the colluders see t_l ones, then for each l we have

$$E[\hat{m}_l] = \frac{2t_l - C}{C}\cdot\frac{1}{L} \tag{51}$$

$$Var[\hat{m}_l] = \frac{L+1}{L^2 M} + \frac{\sigma_n^2}{L}. \tag{52}$$

After hard decoding, the probability of $\hat{b}_l = 1$ conditional to observing t_l ones can be expressed as

$$\begin{aligned}
P_{\hat{b}_l|t_l}(1|t_l) =& Pr\{\hat{m}_l > 0\}\\
=& Q\left(\frac{2t_l - C}{C}\sqrt{\frac{M}{L+1+ML\sigma_n^2}}\right).
\end{aligned} \tag{53}$$

It is easy to observe that $P_{\hat{b}_l|t_l}(1|t_l) = 1 - P_{\hat{b}_l|t_l}(1|C - t_l)$. Based on the above conditional probabilities, the probabilities of $\hat{b}$ conditional to b_k can be expressed as

$$P_{\hat{b}|b_k}(1|1) = \sum_{t=0}^{C-1} \binom{C-1}{t} p^t (1-p)^{C-1-t} P_{\hat{b}|t}(1|t+1) \tag{54}$$

and

$$P_{\hat{b}|b_k}(0|0) = \sum_{t=0}^{C-1} \binom{C-1}{t} (1-p)^t p^{C-1-t} P_{\hat{b}|t}(1|t+1). \tag{55}$$

since the probability of $\hat{b}_l = 0$ conditional to observing t_l zeros is equal to the probability of $\hat{b}_l = 1$ conditional to observing t_l ones, due to the symmetry of the channel.

Using the above expressions, in the case of the accusation score in (25), we obtain

$$\begin{aligned}
\mu_T =& \frac{2L}{\pi} \int_0^1 \left(P_{\hat{b}|b_k}(1|1) + P_{\hat{b}|b_k}(0|0) - 1 \right) dp \\
=& \frac{2L}{\pi} \int_0^1 \Bigg(p^{C-1} P_{\hat{b}|t}(1|C) \\
& - (1-p)^{C-1} \left(1 - P_{\hat{b}|t}(1|C) \right) \\
& + \sum_{t=0}^{C-2} \frac{P_{\hat{b}|t}(1|t+1)}{t+1} \binom{C-1}{t} \\
& \times \frac{\partial p^{t+1} (1-p)^{C-1-t}}{\partial p} \Bigg) dp \\
=& \frac{2L}{C\pi} (1 - 2p_e)
\end{aligned} \tag{56}$$

where $p_e = 1 - P_{\hat{b}|t}(1|C)$ denotes the probability of reading the wrong bit when all colluders agree on the same symbol and we have assumed $\delta = 0$ in the definition of the pdf of p. In the RDM, we have $p_e = 0$. For the variance, we obtain

$$\sigma_T^2 = L \left[1 - \frac{4}{C^2 \pi^2} (1 - 2p_e)^2 \right]. \tag{57}$$

In the case of the accusation score in (29), μ_T and σ_T^2 cannot be expressed in closed form and the expectation over p has to be numerically evaluated.

Experimental results

For the experimental validation of the proposed solutions, we have simulated a system performing client-side embedding on digital images. A dataset of 20 grayscale uncompressed 8 bit images, each having resolution 1024×1024 pixels and representing different subjects, has been used. For each image, a vector $\mathbf{x}$ of 2^{16} components has been obtained by applying a 8×8 discrete cosine transform (DCT) to the image and taking 4 DCT coefficients for each 8×8 block, corresponding to the coefficients between the 7th and 10th positions according to the zig-zag ordering used by JPEG standard.

Encryption has been obtained by adding the elements of the encryption LUT $\mathbf{E}$ to the selected DCT coefficients and reconstructing the images through an inverse block DCT. As in [32], pixel values have been mapped to 9 bits using rounding and modulo 512 operations, so as to avoid wrap around errors after watermarking. In all experiments, the encryption LUT power has been set to $\sigma_E^2 = 10^6$, the LUT size has been set to $T = 2^{16}$, and $R = 4$ LUT entries are added together to encrypt each element. The effect of a fixed point representation of both $\mathbf{E}$ and $\mathbf{D}$ has been assumed negligible.

For each image, 100 independent tests were performed by randomly generating different encryption LUTs, different fingerprints, and different projection matrices $\mathbb{G}$. In each test, we simulated joint decryption and watermarking performed by C different clients, where C ranges from 1 to 100, followed by the averaging attack described in (10). In the attack, the variance of the noise has been set so as to satisfy a prescribed watermark to noise ratio (WNR), defined as $\mathrm{WNR} = 10 \log_{10} \frac{R\sigma_W^2}{\sigma_N^2}$. The performance has been evaluated by measuring the detection rate (DR), defined as the number of tests in which we accuse at least a guilty client over the total number of tests, for each different value of C. Ideally, DR should tend to the theoretical probability of detection of the system.

Random LUTs

For the random LUT solution, we set the fingerprint length to $L = 128$ and we generated the entries of the projection matrices as i.i.d. zero mean variables, according to either a Gaussian or a Bernoulli distribution with values in $1, -1$. Projection matrices have been rescaled so that the power of the watermarking LUT satisfies $\sigma_W^2 = 1$. The threshold used by the detector has been computed according to the Gaussian approximation in (19), assuming a maximum number of users $N_U = 10^6$ and different target values for the probability of false alarm.

In Fig. 2, we show the anticollusion performance of the system for a theoretical probability of false alarm $P_{fa} = 10^{-6}$ and $P_{fa} = 10^{-9}$, assuming that the attack satisfies $\mathrm{WNR} = 0$ dB. The theoretical probability of detection for each collusion set size C has been computed according to (44) and (43) and compared to the measured detection rate obtained using either Gaussian or Bernoulli projection matrices. As can be seen, Gaussian and Bernoulli matrices permit to obtain very

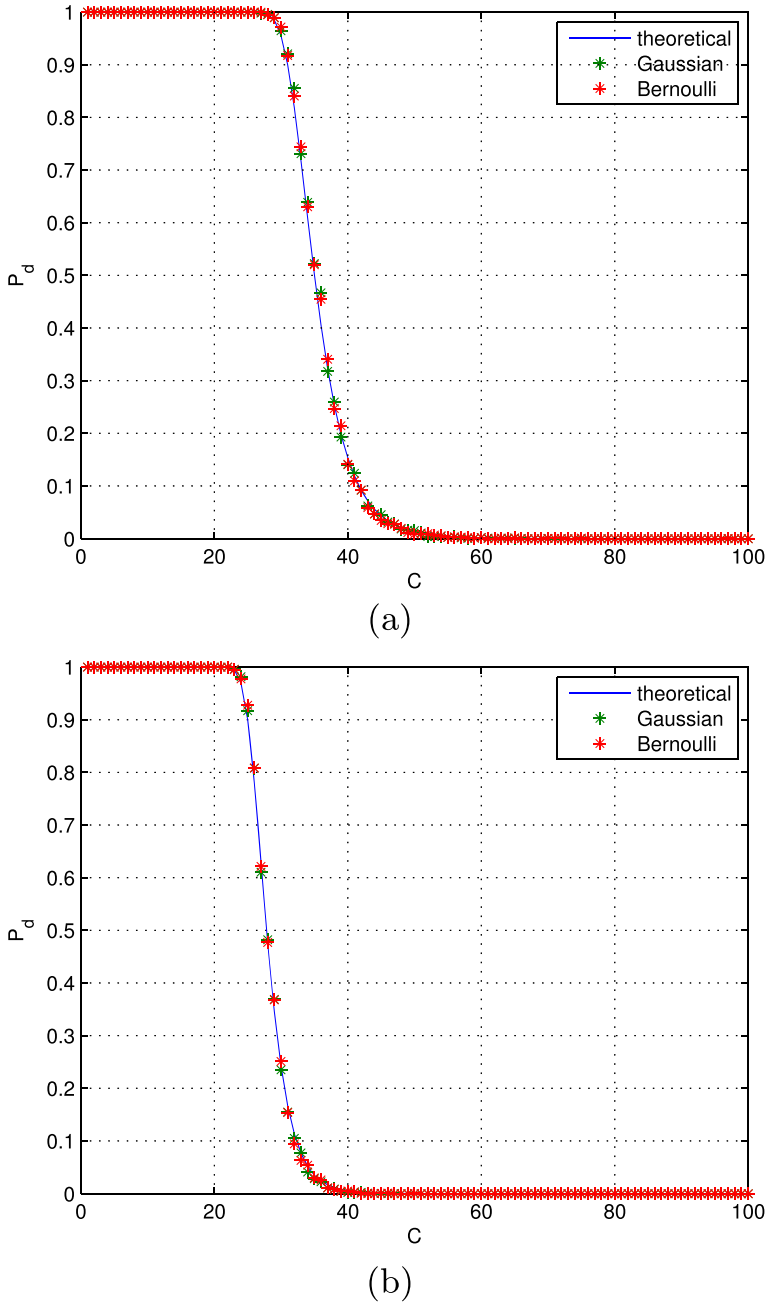


Fig. 2 Performance of random LUT solution for different numbers of colluders at WNR = 0 dB. **a** $P_{fa} = 10^{-6}$; **b** $P_{fa} = 10^{-9}$

similar detection performances. Moreover, the anticollusion performance of the system can be accurately predicted by using the provided theoretical expressions. For the chosen parameters, i.e., $M = 2^{16}$ and $L = 128$, and WNR = 0 dB, the proposed system is able to withstand a coalition of about 32 colluders at $P_{fa} = 10^{-6}$ and 25 colluders at $P_{fa} = 10^{-9}$. These results are similar to the performance of random orthogonal fingerprinting for similar parameters, as reported in [14].

Tardos codes

In the case of Tardos solutions, we similarly generated the entries of the first column of the projection matrices as i.i.d. zero mean variables, according to either a Gaussian or a Bernoulli distribution with values in $1, -1$. Projection matrices have been rescaled so that the power of the watermarking LUT satisfies $\sigma_W^2 = 1$.

For the Tardos solution using the accusation score in (25), simply referred to as Tardos, we set the fingerprint

length to $L = 2^{20}$ and the threshold used by the detector has been computed according to the Gaussian approximation in (28)

For the Tardos solution using the accusation score in (29), referred to as Tardos with interleaving defence (ILD), we set the fingerprint length to $L = 2^{15}$ and the threshold has been computed according to the Gaussian approximation in (36). The variance $\sigma_{T2,I}$ has been numerically estimated by evaluating (35) through Monte Carlo integration, using $P_{\hat{b}_l}(1) = \sum_{t=0}^{C} \binom{C}{t} p^t (1-p)^{C-t} P_{\hat{b}|t}(1|t)$ and $P_{\hat{b}_l}(0) = 1 - P_{\hat{b}_l}(1)$.

In both cases, we assumed a maximum number of users $N_U = 10^6$ and different target values for the probability of false alarm.

In Figs. 3 and 4, we show the anticollusion performance of the Tardos and Tardos ILD solutions, respectively, for a theoretical probability of false alarm $P_{fa} = 10^{-6}$ and $P_{fa} = 10^{-9}$, assuming that the attack satisfies WNR = 0 dB.

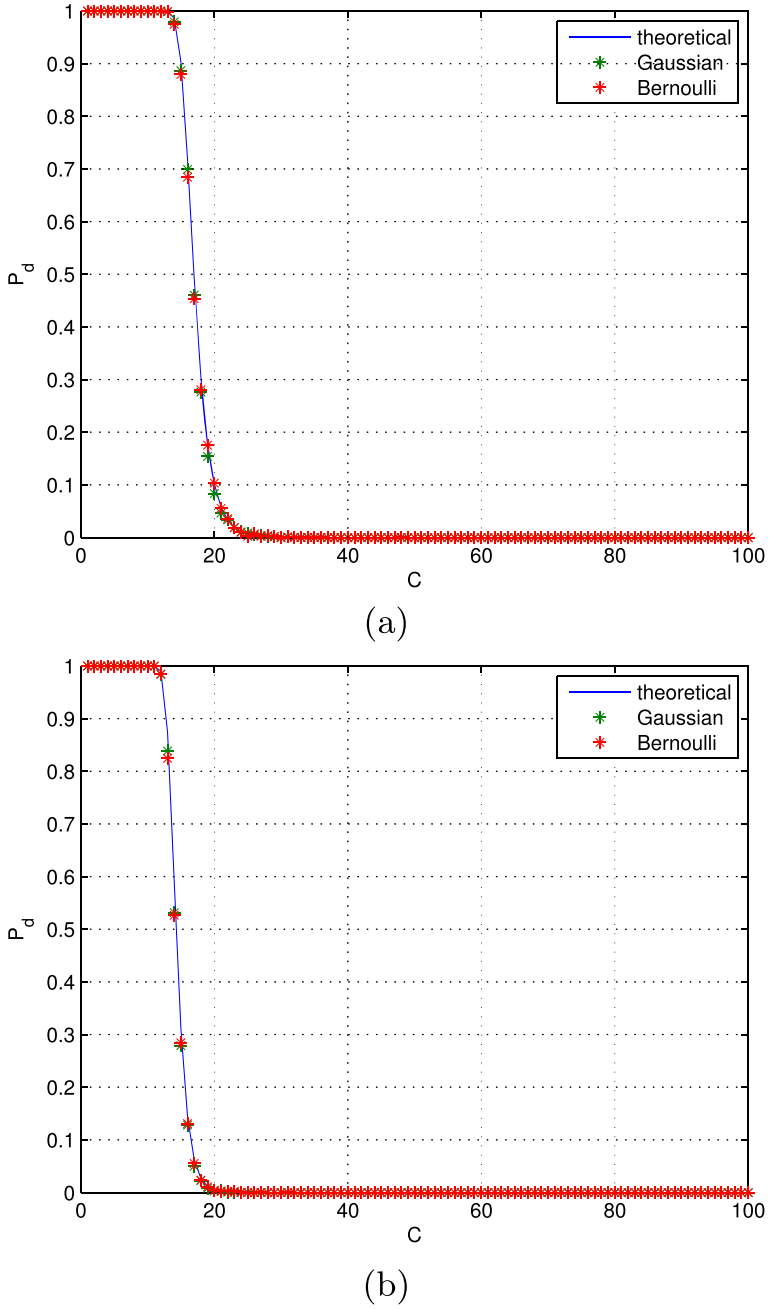
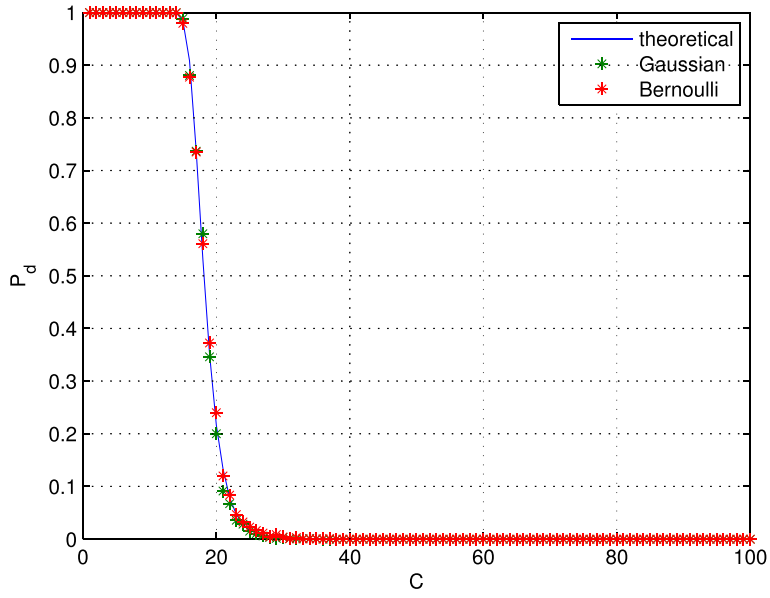
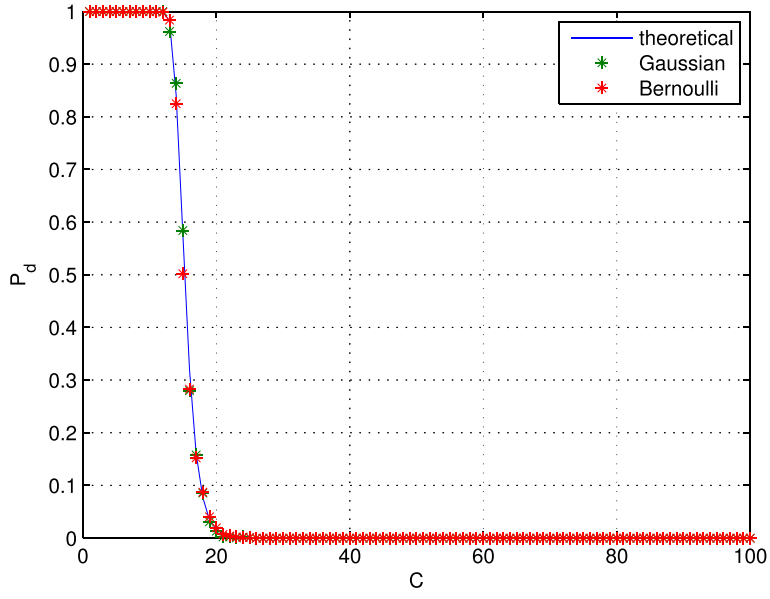


Fig. 3 Performance of Tardos solution, for different numbers of colluders at WNR = 0 dB. **a** $P_{fa} = 10^{-6}$; **b** $P_{fa} = 10^{-9}$



(a)



(b)

Fig. 4 Performance of Tardos ILD solution, for different numbers of colluders at WNR = 0 dB. **a** $P_{fa} = 10^{-6}$; **b** $P_{fa} = 10^{-9}$

The theoretical probability of detection for each collusion set size C has been computed according to (44) and (45) and compared to the measured detection rate obtained using either Gaussian or Bernoulli projection matrices. Also in this case, Gaussian and Bernoulli matrices permit to obtain very similar detection performances and the anticollusion performance of the system can be accurately predicted by using the provided theoretical expressions. At WNR = 0 dB, the Tardos solution with $M = 2^{16}$ and $L = 2^{20}$ is able to withstand a coalition of about 15 0 at $P_{fa} = 10^{-6}$ and 13 colluders at $P_{fa} = 10^{-9}$, whereas the Tardos ILD solution with $M = 2^{16}$ and $L = 2^{15}$ is able to withstand a coalition of about 16 colluders at $P_{fa} = 10^{-6}$ and 14 colluders at $P_{fa} = 10^{-9}$. It is worth noting that the results of the first solution are sensibly worse than the expected performance of Tardos codes on a noiseless

channel, which for $L = 2^{20}$ should be about 87 colluders at $P_{fa} = 10^{-6}$ and $N_U = 10^6$ [19]. In this case, the performance of the code is severely limited by the fact that only $M = 2^{16}$ positions are available to embed the code, which introduces a lot of interference among code bits. The results of ILD solution are in line with the expected performance of Tardos codes for $L = 2^{15}$, which is about 15 colluders at $P_{fa} = 10^{-6}$ and $N_U = 10^6$.

Comparisons

The results of the previous sections show that, for the proposed choice of parameters, the random LUT solution offers better collusion resistance with respect to the embedding of Tardos codes. In this section, we compare the performance of the different solutions for different choices of parameters and under different WNRs of the attack. The comparison is made using the theoretical probability of detection obtained in the previous performance analysis. In the case of the Tardos ILD solution, the quantities μ_T and σ_T^2 have been evaluated through Monte Carlo integration over 10^5 independent realizations of p. For each choice of parameters, we use as performance metric the largest size of the coalition for which the detector can guarantee $P_d > 0.9$.

In Fig. 5, we show a performance comparison between random LUTs and Tardos codes for different values of M. The other parameters are set as in the previous sections, namely WNR $= 0$ dB, $L = 128$ for random LUTs, $L = 2^{20}$ for Tardos codes. For Tardos codes ILD, we tested both $L = 2^{15}$ and $L = 2^{20}$. As to random LUTs, the size of the coalition against which a M-length watermark is secure grows approximately as $C = O(\sqrt{M})$, which is consistent with previous results on random Gaussian fingerprints [13]. For Tardos codes, the behavior depends on the relationship between M and L. For $M < L$, we have the same behavior as random LUTs, even though the size of the coalition against which the system is secure is approximately halved. For $M > L$, the performance depends on the length of the inner Tardos code and is independent of M when $M \gg L$. It is worth noting that the ILD solution performs sensibly better than the baseline Tardos solution for the same length of the inner code.

In Fig. 6, we show a performance comparison for different values of L, setting the other parameters as in the previous section, namely $M = 2^{16}$ and WNR $= 0$ dB. For random LUT and Tardos solutions, choosing a larger L usually increases the anticollusion performance, even though there is a performance threshold for large L. For the Tardos ILD solution, there is an optimal L value, indicating that this solution is much more affected by possible interfence among code bits. For random LUTs, setting $L \approx 10^3$ is sufficient to guarantee the best performance, whereas for Tardos codes, in order to achieve the best attainable performance, $L \gg M$ should be used. For the ILD, the best solution seems to set $L \approx M$.

Finally, in Fig. 7, we show a performance comparison for different WNRs, setting the parameters as $M = 2^{16}$, $L = 128$ for random LUTs, $L = 2^{20}$ and $L = 2^{15}$ for Tardos codes, $L = 2^{15}$ for Tardos codes ILD. At low WNRs,

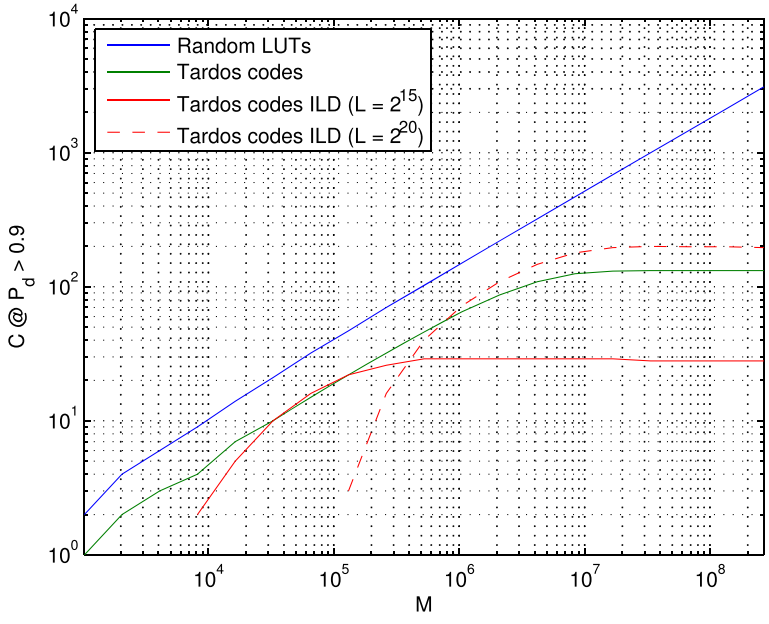


Fig. 5 Performance comparison for different values of M. The other parameters are $L = 128$ for random LUTS, $L = 2^{20}$ for Tardos codes, and WNR $= 0$ dB

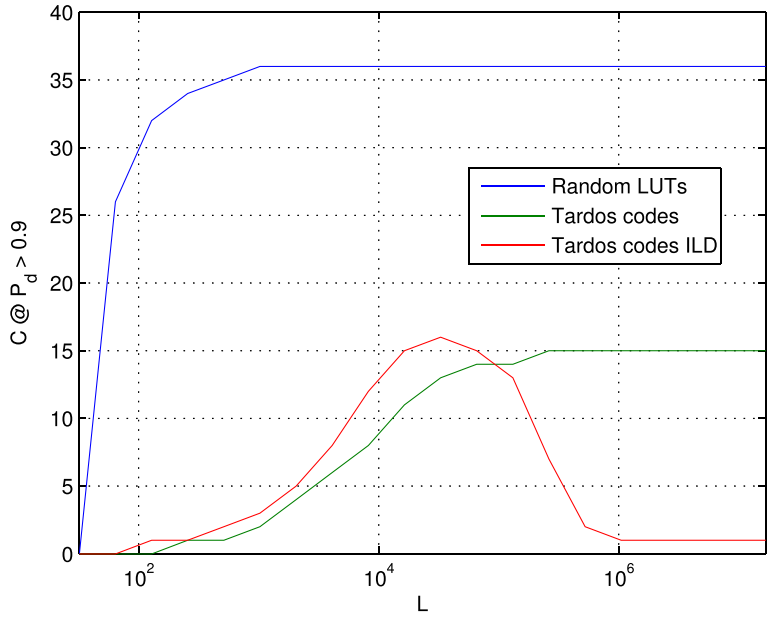


Fig. 6 Performance comparison for different values of L. The other parameters are $M = 2^{16}$, WNR = 0 dB

both systems show a similar behavior: the maximum coalition size grows approximately as $C = O(\sigma_W/\sigma_N)$ and we observe a fixed ratio (approximately equal to two) between the maximum coalition size of random LUTs and Tardos codes. For higher WNRs, the maximum coalition size depends mainly on the other parameters, resulting in a maximum coalition size limit which is independent of the WNR. Interestingly, at high WNRs, random LUTs can withstand a much larger coalition than Tardos codes.

Robustness to JPEG compression

The performance of the two anticollusion solutions in the presence of JPEG compression has been verified experimentally, by using the same parameters as in the

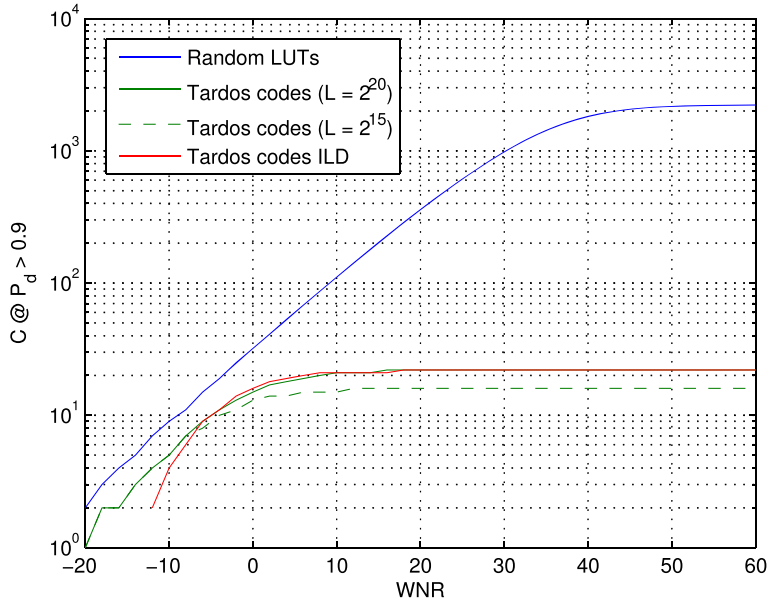


Fig. 7 Performance comparison for different values of WNR. The other parameters are $M = 2^{16}$, $L = 128$ for random LUTS, $L = 2^{20}$ for Tardos codes ILD

previous sections, namely $L = 128$ for random LUTs, $L = 2^{20}$ for Tardos codes, $L = 2^{15}$ for Tardos codes ILD. In Fig. 8, we show the detection rate of random LUT and Tardos solutions, respectively, for $P_{fa} = 10^{-6}$ and a JPEG compression with quality 80. Both techniques are affected by JPEG compression and achieve worse results than in the presence of additive Gaussian noise. It can be observed that JPEG compression at a quality 80 is roughly equivalent to additive noise with WNR $= -5$ dB. This suggests that an embedding space larger than $M = 2^{16}$ should be used to increase the robustness in a realistic scenario including compression after the collusion attack.

Conclusions

Two different anticollusion solutions, specifically tailored for a recently proposed asymmetric fingerprinting protocol based on client-side embedding, have been proposed in this paper. In the first solution, the server randomly encodes a client-owned fingerprint using a different projection matrix for each client. In the second solution, a Tardos code is securely assigned to each client, for example using the protocol in [33], and modulated using a specially designed projection matrix. The performance of both solutions has been analytically derived assuming a correlation-based accusation strategy and an averaging attack by the colluders. For the Tardos solution, two different accusation strategies have been considered, namely, the symmetric accusation score proposed by Škorić et al. and an accusation score optimized for the interleaving attack under the combined digit model. The analytical probability of detecting at least a colluder has been compared to the experimental detection rates, under different number of colluders and different false alarm probability, confirming that the analytical model is indeed very accurate. According to our results, the solution based on independent random projections outperforms the solutions based on binary Tardos codes, under different choices of parameters. Namely, the first solution has a constant performance gain for low watermark-to-noise ratios and appears much more convenient for high watermark-to-noise ratios. The decoder optimized for the interleaving attack is less robust to noisy channels, which is consistent with the fact that its optimality has been demonstrated only for the restricted digit model. Moreover, the solution based on independent random projections requires to manage a much shorter client fingerprint with respect to binary Tardos codes. The same trend has been confirmed by experimental results including a JPEG compression after the averaging attack.

It is worth noting that the proposed schemes have been designed for a scenario that introduces some constraints on the available solutions. Namely, we assume that each client is identified by a binary code, since

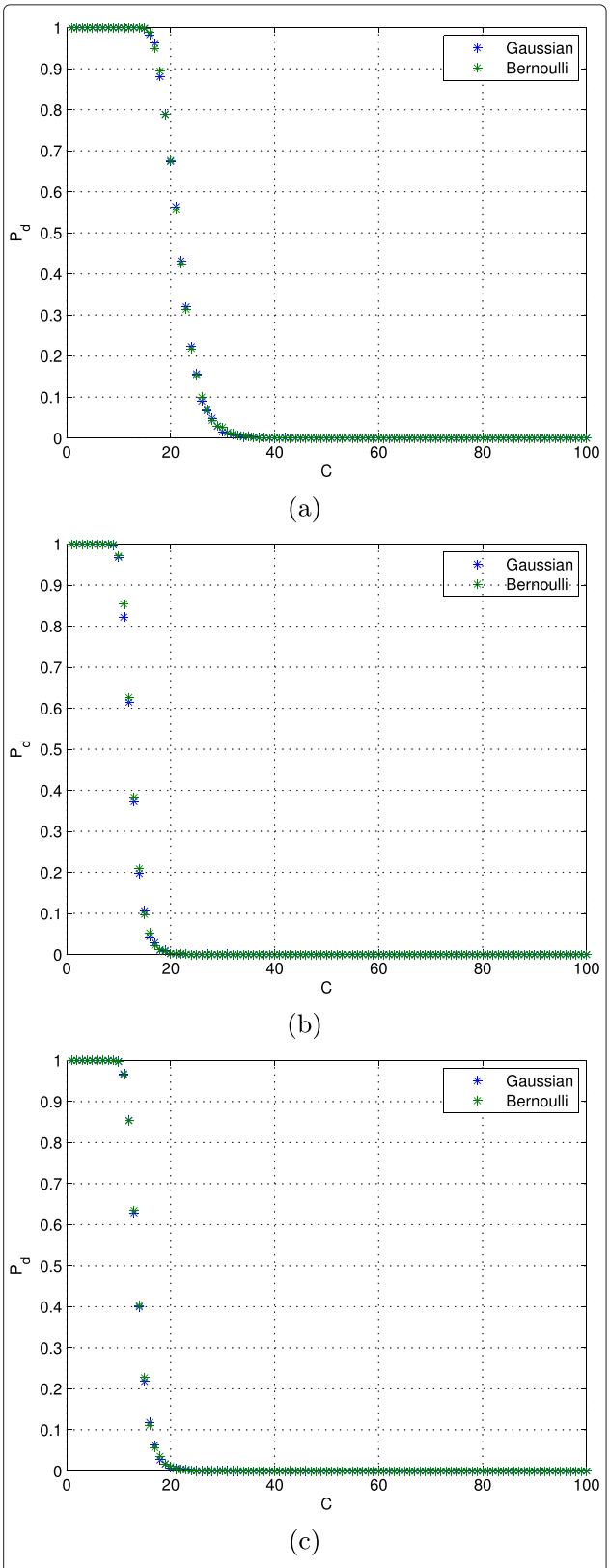


Fig. 8 Performance of different solutions for different numbers of colluders at JPEG quality 80 and $P_{fa} = 10^{-6}$. **a** random LUT; **b** Tardos codes; **c** Tardos codes ILD

we can rely on efficient protocols for securely distributing, in an asymmetric way, an encrypted version of such binary codes [5, 33]. By relaxing these constraints, it would be possible to exploit more powerful non-binary anticollusion codes in the Tardos solutions [19, 21, 26, 27]. Extending the current protocols in order to work with non-binary fingerprints is expected to improve the results and is left for future work.

Competing interests
The authors declare that they have no competing interests.

Author details
[1]Department of Electronics and Telecommunications, Politecnico di Torino, Corso Duca Degli Abruzzi, 24, 10129 Torino, Italy. [2]Department of Information Engineering, University of Florence, Via di S. Marta, 3, 50139 Firenze, Italy.

Received: 15 January 2015 Accepted: 7 August 2015
Published online: 28 August 2015

References

1. M Barni, F Bartolini, *Watermarking Systems Engineering: Enabling Digital Assets Security and Other Applications*. (CRC Press, Boca Raton, FL, USA, 2004)
2. T Bianchi, A Piva, Secure watermarking for multimedia content protection: A review of its benefits and open issues. Signal Process. Mag. IEEE. **30**(2), 87–96 (2013). doi:10.1109/MSP.2012.2228342
3. B Pfitzmann, M Schunter, in *Adv. in Cryptology - EUROCRYPT'96*. LNCS 1070. Asymmetric fingerprinting (Springer Berlin Heidelberg, 1996), pp. 84–95
4. N Memon, P Wong, A buyer-seller watermarking protocol. IEEE Trans. Image Process. **10**(4), 643–649 (2001)
5. M Deng, T Bianchi, A Piva, B Preneel, in *Proceedings of the 11th ACM Workshop on Multimedia and Security*. An efficient buyer-seller watermarking protocol based on composite signal representation (ACM, New York, NY, USA Princeton, New Jersey, USA, 2009), pp. 9–18
6. A Rial, M Deng, T Bianchi, A Piva, B Preneel, A provably secure anonymous buyer-seller watermarking protocol. IEEE Trans. Inform. Forensics Secur. **5**(4), 920–931 (2010)
7. RJ Anderson, C Manifavas, in *Proceedings of the 4th International Workshop on Fast Software Encryption — FSE'97*. Chameleon—a new kind of stream cipher (Springer, London, UK, 1997), pp. 107–113
8. M Celik, A Lemma, S Katzenbeisser, M van der Veen, Look-up table based secure client-side embedding for spread-spectrum watermarks. IEEE Trans. Inform. Forensics Secur. **3**(3), 475–487 (2008)
9. A Piva, T Bianchi, A De Rosa, Secure client-side ST-DM watermark embedding. IEEE Trans. Inform. Forensics Secur. **5**(1), 13–26 (2010)
10. C-Y Lin, P Prangjarote, L-W Kang, W-L Huang, T-H Chen, Joint fingerprinting and decryption with noise-resistant for vector quantization images. Signal Process. **92**(9), 2159–2171 (2012)
11. D Jarnikov, JM Doumen, in *Lecture Notes in Computer Science*, ed. by W Jonker, M Petkovic. Secure Data Management, vol. 6933 (Springer Berlin, Heidelberg, 2011), pp. 101–113
12. IJ Cox, J Kilian, FT Leighton, T Shamoon, Secure spread spectrum watermarking for multimedia. IEEE Trans. Image Process. **6**(12), 1673–1687 (1997). doi:10.1109/83.650120
13. J Kilian, FT Leighton, LR Matheson, TG Shamoon, RE Tarjan, F Zane, in *Proc. of 1998 IEEE International Symposium on Information Theory*. Resistance of digital watermarks to collusive attacks, (1998), p. 271. doi:10.1109/ISIT.1998.708876
14. ZJ Wang, M Wu, HV Zhao, W Trappe, KJR Liu, Anti-collusion forensics of multimedia fingerprinting using orthogonal modulation. IEEE Trans. Signal Process. **14**(6), 804–821 (2005)
15. P Moulin, N Kiyavash, in *2007 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Performance of random fingerprinting codes under arbitrary nonlinear attacks, vol. 2, (2007), pp. 157–160. doi:10.1109/ICASSP.2007.366196
16. D Boneh, J Shaw, Collusion-secure fingerprinting for digital data. IEEE Trans. Inf. Theory. **44**(5), 1897–1905 (1998)
17. G Tardos, in *Proceedings of the Thirty-fifth Annual ACM Symposium on Theory of Computing*. STOC '03. Optimal probabilistic fingerprint codes (ACM, New York, NY, USA, 2003), pp. 116–125
18. B Škorić, TU Vladimirova, M Celik, JC Talstra, Tardos fingerprinting is better than we thought. IEEE Trans. Inf. Theory. **54**(8), 3663–3676 (2008). doi:10.1109/TIT.2008.926307
19. B Škorić, S Katzenbeisser, MU Celik, Symmetric Tardos fingerprinting codes for arbitrary alphabet sizes. Des. Codes Crypt. **46**(2), 137–166 (2008)
20. T Furon, A Guyader, F Cérou, in *Information Hiding*. Lecture Notes in Computer Science, ed. by K Solanki, K Sullivan, and U Madhow. On the design and optimization of Tardos probabilistic fingerprinting codes, vol. 5284 (Springer, Berlin Heidelberg, 2008), pp. 341–356
21. F Xie, T Furon, C Fontaine, in *Proceedings of the 10th ACM Workshop on Multimedia and Security*. MM&Sec '08. On-off keying modulation and Tardos fingerprinting (ACM, New York, NY, USA, 2008), pp. 101–106. doi:10.1145/1411328.1411347
22. K Nuida, S Fujitsu, M Hagiwara, T Kitagawa, H Watanabe, K Ogawa, H Imai, An improvement of discrete Tardos fingerprinting codes. Des. Codes Crypt. **52**(3), 339–362 (2009). doi:10.1007/s10623-009-9285-z
23. E Amiri, G Tardos, in *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*. SODA '09. High rate fingerprinting codes and the fingerprinting capacity (Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2009), pp. 336–345
24. A Charpentier, F Xie, C Fontaine, T Furon. Expectation maximization decoding of Tardos probabilistic fingerprinting code, vol. 7254, (2009), pp. 72540–7254015. doi:10.1117/12.806034
25. B Škorić, S Katzenbeisser, HG Schaathun, MU Celik, Tardos fingerprinting codes in the combined digit model. IEEE Trans. Inf. Forensics Secur. **6**(3), 906–919 (2011). doi:10.1109/TIFS.2011.2116783
26. D Boesten, B Škoric, in *Information Hiding*. Lecture Notes in Computer Science, ed. by T Filler, T Pevný, S Craver, and A Ker. Asymptotic fingerprinting capacity for non-binary alphabets, vol. 6958 (Springer, Berlin, Heidelberg, 2011), pp. 1–13
27. D Boesten, B Škoric, in *Information Hiding*. Lecture Notes in Computer Science, ed. by M Kirchner, D Ghosal. Asymptotic fingerprinting capacity in the combined digit model, vol. 7692 (Springer, Berlin, Heidelberg, 2013), pp. 255–268
28. P Meerwald, T Furon, Toward practical joint decoding of binary Tardos fingerprinting codes. Inf. Forensics Secur. IEEE Trans. **7**(4), 1168–1180 (2012). doi:10.1109/TIFS.2012.2195655
29. J-J Oosterwijk, B Škoric, J Doumen, in *Proceedings of the First ACM Workshop on Information Hiding and Multimedia Security*. IH&MMSec '13. Optimal suspicion functions for Tardos traitor tracing schemes (ACM, New York, NY, USA, 2013), pp. 19–28. doi:10.1145/2482513.2482527
30. T Laarhoven, B de Weger, Optimal symmetric Tardos traitor tracing schemes. Des. Codes Crypt. **71**(1), 83–103 (2014). doi:10.1007/s10623-012-9718-y
31. T Furon, M Desoubeaux, in *2014 IEEE International Workshop on Information Forensics and Security (WIFS)*. Tardos codes for real, (2014), pp. 24–29. doi:10.1109/WIFS.2014.7084298
32. T Bianchi, A Piva, TTP-free asymmetric fingerprinting based on client side embedding. IEEE Trans. Inf. Forensics Secur. **9**(10), 1557–1568 (2014)
33. A Charpentier, C Fontaine, T Furon, I Cox, in *Proceedings of the 13th International Conference on Information Hiding*. IH'11. An asymmetric fingerprinting scheme based on Tardos codes (Springer, Berlin, Heidelberg, 2011), pp. 43–58
34. S Pehlivanoglu, in *Proceedings of the First ACM Workshop on Information Hiding and Multimedia Security*. IH& MMSec '13. An asymmetric fingerprinting code for collusion-resistant buyer-seller watermarking (ACM, New York, NY, USA, 2013), pp. 35–44
35. S Katzenbeisser, B Škorić, M Celik, A-R Sadeghi, in *Information Hiding*. Lecture Notes in Computer Science, ed. by T Furon, F Cayre, G Doërr, and P Bas. Combining Tardos fingerprinting codes and fingercasting, vol. 4567 (Springer, Berlin Heidelberg, 2007), pp. 294–310
36. WB Johnson, J Lindenstrauss, Extensions of Lipschitz mappings into a Hilbert space. Contemp. Math. **26**, 189–206 (1984)
37. DL Donoho, Compressed sensing. Inf Theory IEEE Trans. **52**(4), 1289–1306 (2006)
38. MU Celik, AN Lemma, S Katzenbeisser, M van der Veen, in *IEEE International Conference on Acoustics, Speech and Signal Processing, 2007. ICASSP 2007.*

Secure embedding of spread spectrum watermarks using look-up-tables, vol. 2, (2007), pp. 153–156. doi:10.1109/ICASSP.2007.366195
39. T Bianchi, A Piva, M Barni, On the implementation of the discrete Fourier transform in the encrypted domain. IEEE Trans. Inf. Forensics Secur. **4**(1), 86–97 (2009)
40. Y-W Huang, P Moulin, On the saddle-point solution and the large-coalition asymptotics of fingerprinting games. IEEE Trans. Inf. Forensics Secur. **7**(1), 160–175 (2012). doi:10.1109/TIFS.2011.2168212
41. A Jarrous, B Pinkas, in *Applied Cryptography and Network Security*. Lecture Notes in Computer Science, ed. by M Abdalla, D Pointcheval, P-A Fouque, and D Vergnaud. Secure hamming distance based computation and its applications, vol. 5536 (Springer, Berlin Heidelberg, 2009), pp. 107–124